UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/634,304	08/04/2003	Paulo Pacheco	MWS-030	7963

959 7590 05/15/2007
LAHIVE & COCKFIELD, LLP
ONE POST OFFICE SQUARE
BOSTON, MA 02109-2127

EXAMINER	
WANG, RONGFA PHILIP	

ART UNIT	PAPER NUMBER
2191	

MAIL DATE	DELIVERY MODE
05/15/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	**Application No.** 10/634,304	**Applicant(s)** PACHECO ET AL.	
	Examiner Philip Wang	**Art Unit** 2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *21 March 2007*.

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-47* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-47* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) □ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

1. This office action is in response to amendment filed on 3/21/2007.

2. The objection to the specification is withdrawn in view of the Applicant's amendment to the specification.

3. The objection to the claim 40 is withdrawn in view of the Applicant's amendment to the claims.

4. The 35 USC § 112, second paragraph rejections of claims 10-13, and 17-18 are withdrawn in view of the Applicant's amendment.

5. Per Applicant's request, claims 10, 13, 17, and 18 have been amended.

6. Claims 1-47 remain pending.

Claim Rejections - 35 USC § 112

The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

7. Claim 35 is rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention.

Claim 35 recites the limitation of inheriting the state of the object from an abstract class. On page 11, 3rd paragraph, for example, "...The parent class is not suitable for instantiation and used to abstract out incomplete set of features…" In essence, and abstract class cannot be used for instantiation, so it is impossible to have an object inheriting the state from an abstract class.

Claim Rejections - 35 USC § 102

> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

1. Claims 1-3, 5-7, 21-23, 26-33, 36-41, and 44-47 are rejected under 35 U.S.C. 102(b) as being anticipated by Denk et al. (US PGPub. No. 2001/0025292).

As per claim 1,

Denk et al. disclose

A method for implementing and using a filter object which generates an output in response to an input of the filter object, wherein the output of the filter object depends on the input and a state of the filter object, wherein the state of the filter object includes a minimum amount of information necessary to determine the output of the filter object, the method comprising:

- providing the filter object, the filter object being represented by equations performed to generate the output in response to the input of the filter object, the equations including the state of the filter object ([0031], for example, line 8-9, "…can be represented by the equation…"; See Fig. 3, [0045]); and retaining the state of the filter object, wherein the filter object is implemented and used in a first dynamically typed text-based programming environment ([0070], "The Matlab®…" where it shows a dynamically types text-based programming environment is used to implement such filter.).

As per claim 2,

the rejection of claim 1 is incorporated; further Denk et al. disclose

- the filter object retains a final value of the state obtained as a result of processing the input of the filter object (See Fig. 3, [0045]).

As per claim 3,

the rejection of claim 2 is incorporated; further Denk et al. disclose

- the final value of the state retained in the filter object is used as an initial value of the state for processing the input of the system (See Fig. 3, [0045]).

As per claim 5,

the rejection of claim 1 is incorporated; further Denk et al. disclose

- the step of presetting the state of the filter object retained in the filter object ([0108], line 13, "...preset the threshold value...").

As per claim 6,

the rejection of claim 1 is incorporated; further Denk et al. disclose

- the output of the filter object is determined depending on a present input and a previous input of the filter object ([0108], line 13, "...preset the threshold value..."; [0109], "...responsive to a predetermined characteristic of one more of signal X...").

As per claim 7,

the rejection of claim 6 is incorporated; further Denk et al. disclose

- the state of the filter object contains information about the previous input of the filter object(See Fig. 3, [0045]).

As per claim 21,

Denk et al. disclose

In a computer-implemented system, a method for generating an output of the system in response to an input of the system, the method comprising the steps of:

- specifying a state of the system that includes a minimum amount of information that is necessary to determine the output of the system; retaining the state of the system in a memory; providing to the system the state of the system retained in the memory; and determining the output of the system depending on the input and a state of the system([0053], "...Initially, the rounding operand is assigned...in a manner consistent with the reduced precision desired for output signal..."), wherein the method is implemented in a dynamically typed text-based programming environment([0070], "The Matlab®..." where it shows a dynamically types text-based programming environment is used to implement such filter.).

As per claim 22,

the rejection of claim 21 is incorporated;

further Denk et al. disclose

- the step of specifying equations that the system performs to generate the output of the system from the input and the state of the system([0031], for example, line 8-9, "...can be represented by the equation..."; Fig. 3).

As per claim 23,

the rejection of claim 21 is incorporated;

further Denk et al. disclose

- the step of controlling the state of the system retained in the memory ([0108], "…preset the threshold value…selectable…").

As per claim 26,

the rejection of claim 21 is incorporated;

- It is rejected for the same reason for the rejection claim 2.

As per claim 27,

the rejection of claim 21 is incorporated;

further Denk et al. disclose

- the state of the system provided to the system includes an initial state of the system for processing the input of the system ([0082], line 16-17, "…In this case, the rounding operand would be initially assigned…").

As per claim 28,

Denk et al. disclose

A computer readable medium holding instructions executable in a computer that provides a dynamically typed text-based programming environment ([0070], "The Matlab®…" where it shows a dynamically types text-based programming environment is used to implement such filter.), wherein the computer generates an output of an object in response to an input of the object, comprising:

- providing a class, the object being an instance of the class ([0071]-[0076] for Matlab® code);

- specifying a state of the object that includes a minimum amount of information that is necessary to determine the output of the system, the state being a property of the object; and determining the output of the object depending on the input and the state of the system([0053], "...Initially, the rounding operand is assigned...in a manner consistent with the reduced precision desired for output signal...").

As per claim 29,

the rejection of claim 28 is incorporated;

further Denk et al. disclose

- the step of instantiating the object from the class([0072]-[0076], for code example.).

As per claim 30,

the rejection of claim 28 is incorporated;

further Denk et al. disclose

- the object includes an adaptive filter object ([0028], "... adaptive filter...").

As per claim 31,

the rejection of claim 30 is incorporated;

further Denk et al. disclose

- the adaptive filter object includes an adapting algorithm that the adaptive filter performs([0031], "...can be represented by the equation...").

As per claim 32,

the rejection of claim 28 is incorporated;

further Denk et al. disclose

- the object includes a discrete time filter object([0034], line 9-12, "... signal sample...discrete time sequence of signal samples...").

As per claim 33,

the rejection of claim 28 is incorporated;

further Denk et al. disclose

- the step of controlling properties of the object including the state of the object([0108], "...preset the threshold value...selectable...").

As per claim 36,

the rejection of claim 28 is incorporated;

further Denk et al. disclose

- the step of providing the class with methods which operate on the object of the class([0072]-[0075]).

As per claim 37,

A system for implementing a filter object which generates an output in response to an input of the filter object, wherein the output of the filter object depends on the input and a state of the filter object, wherein the state of the filter object includes a minimum amount of information necessary to determine the output of the filter object, the method comprising:

- a memory for retaining the state of the filter object ([0109], line 8, "Memory 1138 can be used...."); and
- a state equation processing unit for generating a new state of the filter object based on the state of the filter object retained in the memory and the input of the filter object([0031], for example, line 8-9, "...can be represented by the equation..."; See Fig. 3, [0045]); wherein the filter object is implemented and used in a first dynamically typed text-based programming environment([0070], "The Matlab®..." where it shows a dynamically types text-based programming environment is used to implement such filter.).

As per claim 38,

the rejection of claim 37 is incorporated;

further Denk et al. disclose

- the memory retains the new state of the filter object(See Fig. 3, [0045]).

As per claim 39,

the rejection of claim 38 is incorporated;

further Denk et al. disclose

- the new state retained in the memory is used as a state of the filter object in processing next input of the filter object(See Fig. 3, [0045]).

As per claim 40,

the rejection of claim 37 is incorporated;

further Denk et al. disclose

- an output equation processing unit for generating the output of the filter object based on the state of the filter object retained in the memory and the input of the filter object([0031], for example, line 8-9, "...can be represented by the equation...";).

As per claim 41,

the rejection of claim 37 is incorporated;

further Denk et al. disclose

- the state of the filter object contains information about the previous input of the filter object (See Fig. 3, [0045]).

As per claim 44,

the rejection of claim 1 is incorporated; further Denk et al. disclose

- the filter object operates on a sample-by-sample, block-by-block or frame-by-frame basis([0034], line 9-12, "... signal sample...discrete time sequence of signal samples...").

As per claims 45, 46, and 47,

- they are rejected for the same reason for the rejection of claim 44.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claims 4, 24, 25, 34, 42, and 43 are rejected under 35 U.S.C. 103(a) as being unpatentable over Denk et al. (US PGPub. No. 2001/0025292) in view of Gay (US Patent No. 5,677,951).

As per claim 4,

the rejection of claim 1 is incorporated;

Denk et al. do not specifically disclose

- the step of resetting the state of the filter object retained in the filter object.

However, Gay disclose

- the step of resetting the state of the filter object retained in the filter object (c3, line 5-26, specifically, line 11, "…a restart signal…").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of Gay into the teachings Denk et al. to include the step of resetting the state of the filter object retained in the filter object. The modification would be obvious to one of ordinary skill in the art to want to allow the control of the filer by a user as suggested by Gay (c3: 13-15).

As per claims 24, 34 and 42,

- they are rejected for the same reason as claim 4.

As per claim 25,

the rejection of claim 23 is incorporated;

Denk et al. do not specifically disclose

- the state of the system retained in the memory is set to a particular value entered by a user.

However, Gay disclose

- the state of the system retained in the memory is set to a particular value entered by a user(c3, line 5-26, specifically, line 11, "...a restart signal...").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of Gay into the teachings Denk et al. to include the state of the system retained in the memory is set to a particular value entered by a user. The modification would be obvious to one of ordinary skill in the art to want to allow the control of the filer by a user as suggested by Gay (c3: 13-15).

As per claim 43,

- It is rejected for the same reason as claim 25.

3. Claims 8, 19, and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Denk et al. (US PGPub. No. 2001/0025292) in view of "AutoCode Solutions" (herein AutoCode, http://web.archive.org/web/20021120051701/http://www.filter-solutions.com/, dated 2002).

As per claim 8,

the rejection of claim 1 is incorporated;

Denk et al. do not specifically disclose

- the filter object is utilized to generate code to implement a corresponding filter algorithm separate from the filter object implementation.

However, Autocode discloses

- the filter object is utilized to generate code to implement a corresponding filter algorithm separate from the filter object implementation (page 1, 1st para., "...provides the capability to generate C code for your digital filter...").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of Autocode into the teachings of Denk et al. to include the filter object is utilized to generate code to implement a corresponding filter algorithm separate from the filter object implementation. The modification would be obvious to one of ordinary skill in the art to want to use the code in standard compatible compiler as suggested by Autocode (p. 1, 1st para.).

As per claim 19,

the rejection of claim 8 is incorporated;

further Autocode discloses

- the code is a high-level programming language (page 1, 1st para., "...provides the capability to generate C code for your digital filter...").

As per claim 20,

the rejection of claim 8 is incorporated; further Autocode disclose

- the code is a low-level machine or assembly language(page 1, 1st para.,"...The code is compatible with any standard C or C++ compiler".).

4. Claims 9-18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Denk et al. (US PGPub. No. 2001/0025292) in view of "AutoCode Solutions" (herein AutoCode, http://web.archive.org/web/20021120051701/http://www.filter-solutions.com/, dated 2002), and further in view of "Digital Filter Solutions", (herein DFS, http://web.archive.org/web/20021120051701/http://www.filter-solutions.com/, dated 2002).

As per claim 9,

the rejection of claim 1 is incorporated;

Denk et al. do not specifically disclose

- the filter object is utilized to generate code to implement a corresponding test bench or filter analysis.

However, Autocode discloses

- the filter object is utilized to generate code `(page 1, 1st para., "…provides the capability to generate C code for your digital filter…")`.

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of Autocode into the teachings of Denk et al. to include the filter object is utilized to generate code to implement a corresponding filter algorithm separate from the filter object implementation. The modification would be obvious to one of ordinary skill in the art to want to use the code in standard compatible compiler as suggested by Autocode (p. 1, 1st para.).

Both Denk et al. and Autocode do not disclose

- implement a corresponding test bench or filter analysis.

However, DFS discloses

- implement a corresponding test bench or filter analysis (p. 9, "Precision and Quantization", "...provide digital filter test...").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of DFS into the teachings of both Denk et al. and Autocode to include "implement a corresponding test bench or filter analysis". The modification would be obvious to one of ordinary skill in the art to want to determine if a digital filter will execute properly on the target environment as suggested by DFS (p. 9, 2nd para., line 4, "..determine if your digital filer will execute properly...).

As per claim 10,

the rejection of claim 8 is incorporated;

Denk et al. do not disclose

- the generated code can be executed, directly or via a suitable compilation process,
- However, Autocode dislcoses
- the generated code can be executed, directly or via a suitable compilation process, (p. 1, 1st para., "The code is compatible with any standard C or C++ compiler."),

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of Autocode into the teachings of Denk et al. to include the generated code can be executed, directly or via a suitable compilation process, on the host machine. The modification would be obvious to one of ordinary skill in the art to want to use the code in standard compatible compiler as suggested by Autocode (p. 1, 1st para.).

Both Denk et al. and Autocode do not disclose

- outside the context of a simulation environment on which the filter executes;

However, DFS discloses

- outside the context of a simulation environment on which the filter executes (p. 9, "Precision and Quantization", "...the digital time response from the Filter Solution Filter control panel...provides digital filters and simulation features...on your target processor").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of DFS into the teachings of Both Denk et al. and Autocode to include "outside the context of the simulation environment on which the filter executes". The modification would be obvious to one of ordinary skill in the art to want to determine if a digital filter will execute properly on the target environment as suggested by DFS (p. 9, 2nd para., line 4, "..determine if your digital filer will execute properly...).

As per claims 11 and 12,

the rejection of claim 10 is incorporated;

further Autocode disclose

- the generated code is a textual language/ a graphical description language (see page 2 code list in text. The examiner asserts that C can be used a graphical description language).

As per claim 13,

the rejection of claim 8 is incorporated;

Denk et al. do not disclose

- the generated code can be executed, directly or via a suitable compilation process, However, Autocode dislcoses

- the generated code can be executed, directly or via a suitable compilation process, (p. 1, 1st para., "The code is compatible with any standard C or C++ compiler."),

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of Autocode into the teachings of Denk et al. to include the generated code can be executed, directly or via a suitable compilation process, on the host machine. The modification would be obvious to one of ordinary skill in the art to want to use the code in standard compatible compiler as suggested by Autocode (p. 1, 1st para.).

Both Denk et al. and Autocode do not dislcose

However, DFS discloses

- on the host machine, within the context of a simulation environment on which the filter executes (p. 9, "Precision and Quantization", "...the digital time response from the Filter Solution Filter control panel...provides digital filters and simulation features...").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of DFS into the teachings of Both Denk et al. and Autocode to include "on the host machine, within the context of the simulation environment on which the filter executes". The modification would be obvious to one of ordinary skill in the art to want to determine if a digital filter will execute properly on the target environment as suggested by DFS (p. 9, 2nd para., line 4, "..determine if your digital filer will execute properly...).

As per claim 14,

the rejection of claim 8 is incorporated;

Denk et al. do not disclose

- the generated code can be executed, directly or via a suitable compilation process,

However, Autocode dislcoses

- the generated code can be executed, directly or via a suitable compilation process, (p. 1, 1st para., "The code is compatible with any standard C or C++ compiler."),

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of Autocode into the teachings of Denk et al. to include the generated code can be executed, directly or via a suitable compilation process, on the host machine. The modification would be obvious to one of ordinary skill in the art to want to use the code in standard compatible compiler as suggested by Autocode (p. 1, 1st para.).

Both Denk et al. and Autocode do not disclose

- an embedded system implementation.

However, DFS discloses

further DFS disclose

- an embedded system implementation (p. 9, 2nd para., line 4, "..determine if your digital filer will execute properly on your target system.").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of DFS into the teachings of Both Denk et al. and Autocode to include "on the host machine, within the context of the simulation environment on which the filter executes". The modification would be obvious to one of ordinary skill in the art to want to determine if a digital filter will execute properly on the target environment as suggested by DFS (p. 9, 2nd para., line 4, "..determine if your digital filer will execute properly…).

As per claim 15,

the rejection of claim 14 is incorporated;

- see reason for rejection of claim 11.

As per claim 16,

the rejection of claim 14 is incorporated;

- see reason for rejection of claim 12.

As per claim 17,

the rejection of claim 14 is incorporated;

further DFS discloses

- the generated code is suitable for use with a software implementation, including use on a general purpose processor, a digital signal processor, or other programmable compute architecture (p. 9, "Precision and Quantization", "... will execute properly on your target processor...").

As per claim 18,

the rejection of claim 14 is incorporated; further Denk et al. disclose

- the generated code is suitable for use with a hardware implementation, including use with t least one of a Field Programmable Gate Array (FPGA), Complex Programmable Logic Device (CPLD), and Application Specific Integrated Circuit (ASIC) device, the generate code being written in hardware description language. ([0061], "...Verilog HDL...").

Response to Arguments

In the remark,

Applicant argues:

1) Applicant argues that claim 35 is supported by the specification, where claim 35 claims -- "The medium of claim 28, further comprising the step of inheriting the state of the object from an abstract class."

Examiner's response:

1) The Application indicates support for this limitation can be found at page 11, paragraph 3; page 12, paragraph 1. Upon reviewing the above section, the examiner does not consider the disclosure support the claim limitation.

Per claim language of claim 35, "the step of inheriting the state of the object from an abstract class". The examiner interprets "the state of the object" refers to the value(s) that represents the state the object. There is distinction between value(s) being assigned to a defined data structure and the definition of a defined data structure itself. Fig. 4B shows two abstract classes (BaseFilter and AbstractFilter) and one class (DFILT). The Applicant recites, for example, "The DFILT class 430 inherits from the AbstractFilter, the FilterStructure 421, States 423 and ResetStates 425 properties along with other properties that define the filter." What are being inherited here are the properties that define the filter, not the value(s) being assigned. What the DFLIT inherits from the AbstractFilter is the definition of data structure, such as "States", not the value(s) being assigned to it. When the DFILT is instantiated, it uses the definition in its parent abstract class to create a data structure called "States" to store the value(s) of the state of the object. DFILT does not inherit the value(s) assigned to "States" from the abstract class

AbstractFilter, instead, it inherits the definition (or declaration) of the state of the object from an abstract class.

Applicant argues:

2) Denk does not disclose "retaining the state of the filter object," (per claims 1, 2-3, 5-20)

Examiner's response:

2) On page 2, lines 27-30 of the specification, "The state of the system is specified to include a minimum amount of information that is necessary to determine the output of the system. The state of the system is retained in a memory and provided to the system to determine the output of the system." Therefore, the examiner interprets "retaining the state of the filter object" as storing information necessary to determine the output of the system in a memory.

Denk shows, for example, in Fig. 3, a filter with necessary information, such as data source, and bias (α) to determine the output of the system. In a computing system, in order for data to be processes, data must be stored in memory. Therefore, information necessary to determine the output of the system is retained. The language as is presented in the claims does not appear to be able differentiate from the above interpretation.

Applicant argues:

3) Denk does not disclose "retaining the state of the filter object,", and "determining the output of the system depending on a state of the system". (per claims 21-23, 26-27, 28-33, 36)

Examiner's response:

3) Based on reasons provided in examiner's response 2, data are retained in memory to determine output. [0053] describes input and other data are provided an stored in memory to determine the output.

Applicant argues:

4) Denk does not show "retain the state of the system and does not include a filter object" (per claim 37-47)

Examiner's response:

4) [0031] and Fig. 1 shows a filter that takes data input and determines data output. The examiner considers an object taking data input, process it, and produces data output as a filter. See examiner's response 2 for reasons supporting retaining the state of the system.

Applicant argues:

5) Denk and Gay fail to disclose "retaining the state of the filter object, wherein the filter object is implemented and used in a first dynamically typed text-based programming environment."

Examiner's response:

5) Regarding arguments related to retaining the state of the filter object, please refer to examiner's response in 2. [0070] of Denk shows a Matlab® environment, which is a dynamically typed text-based programming environment.

Applicant argues:

6) The examiner takes the position that a restart signal is not the equivalent to the state of a filter object. (per claim 4)

Examiner's response:

6) The examiner does not hold the opinion of a restart signal is the equivalent to the state of a filter. Instead, use of a restart signal resets the filter object.

Conclusion

THIS ACTION IS MADE FINAL. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Philip Wang whose telephone number is 571-272-5934. The examiner can normally be reached on Mon - Fri 8:00 - 4:00PM. Any inquiry of general nature or relating to the status of this application should be directed to the TC2100 Group receptionist: 571-272-2100.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Zhen can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

WEI ZHEN
SUPERVISORY PATENT EXAMINER